

## Aplikasi Jaringan Saraf Tiruan Sebagai Penterjemah Karakter Braille Ke Bentuk Abjad

Anggara Ismawan, Setiawardhana, Dwi Kurnia Basuki

Program Studi Teknik Komputer

Politeknik Elektronika Negeri Surabaya – Institut Teknologi Sepuluh Nopember  
Kampus ITS Keputih Sukolilo Surabaya 60111 Telp. 031-5947280, Fax 031-5946114

### Abstrak

*Sistem pengkonversi karakter Braille merupakan sistem yang dapat melakukan penerjemahan karakter Braille ke bentuk abjad. Penelitian ini bertujuan membuat aplikasi pendeteksian dan penterjemah karakter Braille menggunakan Backpropagation. Pendeteksian citra Braille dilakukan dengan ekstraksi fitur Braille menggunakan threshold. Dengan metode Backpropagation sistem dapat menganalisa informasi berupa citra, dan dikonversi ke dalam kode huruf. Hasil penelitian ini adalah sistem dapat mengenali karakter Braille ke bentuk abjad secara offline menggunakan Backpropagation dengan pattern sebanyak 81 data diperoleh error goal sebesar 0.001 dan training berhenti pada epoch ke 651.*

Keywords: braille, threshold, backpropagation.

### 1. Pendahuluan

Saat ini banyak peneliti yang mengembangkan sistem komputer untuk secara otomatis dapat mengenali berbagai objek misalnya Braille. Braille adalah sejenis sistem tulisan sentuh yang digunakan oleh orang buta. Sistem ini sangat membantu orang buta untuk proses berkomunikasi dalam bentuk tulisan. Setiap sel Braille terdiri dari enam titik timbul, tiga baris dengan dua titik. Thiang[4] mencoba mengembangkan sistem pengenalan karakter Braille menggunakan *Hamming network* yang merupakan salah satu model struktur dari *artificial neural network*. Lisa Wong, Waleed Abdulla dan Stephan Hussman[9] mengembangkan metode jaringan saraf tiruan untuk pengenalan karakter Braille menggunakan *image processing* dari proses pengambilan sample Braille menggunakan mesin *scanner*.

Pada penelitian pengenalan karakter Braille dan *thresholding*, Y.M. Alginahi[8] menggunakan metode *MultiLayer Perceptron Neural Network (MLP NN)* dengan sistem *Optical Character Recognition (OCR)* yang dapat menerjemahkan suatu *image* ke dalam bentuk teks.

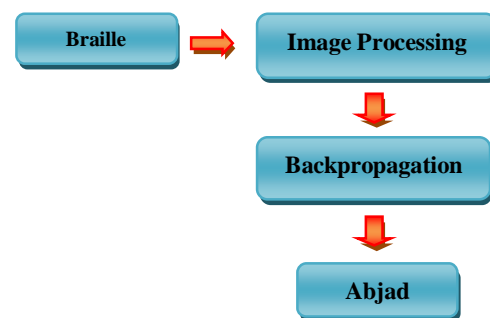
Pada penelitian ini akan dikembangkan sistem penterjemah karakter Braille menggunakan metode Jaringan Saraf Tiruan perambatan – balik (*JST backpropagation*). Secara garis besar proses penelitian ini adalah pengenalan citra karakter Braille menggunakan *image processing* dengan cara mengkombinasikan antara ekstraksi fitur dan jaringan saraf tiruan yang dikembangkan menggunakan metode *Backpropagation*.

### 2. Pembahasan

Pada bagian ini akan dibahas desain dan implementasi sistem dengan metode yang diusulkan sampai dengan analisa hasil pengujian sistem secara keseluruhan.

#### 2.1. Perancangan Sistem

Penelitian ini bertujuan untuk mendesain dan membuat sistem pengenalan karakter Braille menggunakan *image processing* dengan jaringan saraf tiruan menggunakan metode *Backpropagation*. Gambar 1 menunjukkan diagram rancangan sistem pengenalan karakter Braille secara keseluruhan :

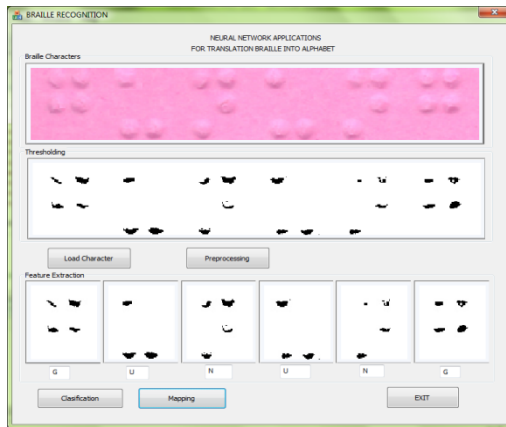


**Gambar 1.** Sistem Pengenalan Karakter Braille

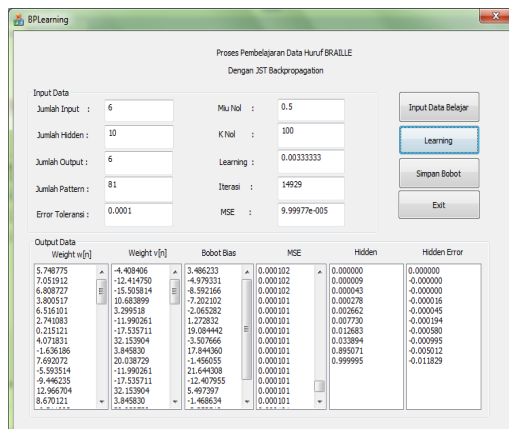
Karakter Braille diambil dari proses *scanning* dengan mesin *scanner* kemudian dilakukan *image processing* yaitu melakukan *thresholding* Braille, sehingga diperoleh ekstraksi fitur dari Braille. Informasi yang diperoleh dari hasil *image processing* adalah berupa data biner yaitu 0 dan 1. Informasi tersebut digunakan

sebagai nilai masukan ke *Backpropagation*. Model dari sistem pengenalan dengan *Backpropagation* yang dirancang mempunyai 6 buah data masukan dan 5 buah pasangan data sebagai target.

Aplikasi sistem ini dibuat dengan menggunakan visual studio 2008 seperti pada gambar 2 dan gambar 3.



**Gambar 2.** Form Aplikasi *Image Processing*









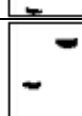
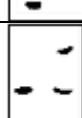
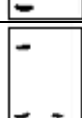

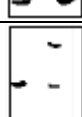


**Gambar 3.** Form Aplikasi *Backpropagation*


## 2.2. Pengambilan Fitur Karakter Braille

Penelitian ini bertujuan untuk mengetahui fitur dari karakter Braille hasil proses *image processing* yaitu *thresholding* pada semua karakter Braille sebanyak 26 data. Setiap fitur karakter Braille memiliki data fitur yang berbeda, pada table 1 menunjukkan fitur karakter Braille hasil *thresholding* :

**Tabel 1.** Pengambilan Fitur Karakter Braille

No.	Thresholding	Abjad	Fitur Braille
1		a	100000
2		b	110000
3		c	100100
4		d	100110
5		e	100010
6		f	110100
7		g	110110
8		h	100110
9		i	010100
10		j	010110
11		k	101000
12		l	111000

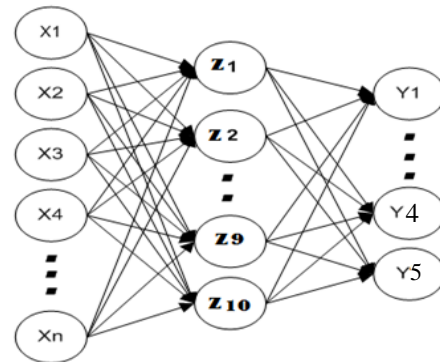
13		m	101100
14		n	101110
15		o	101010
16		p	111100
17		q	111110
18		r	111010
19		s	011100
20		t	011110
21		u	101001
22		v	111001
23		w	010111
24		x	101101
25		y	101111

26		z	101011
----	---	---	--------

### 2.3. Pelatihan Data Karakter Braille

Hasil data yang diperoleh dari ekstraksi fitur karakter Braille akan memasuki fase pelatihan dan selanjutnya dipelajari dengan sistem JST *Backpropagation*. Pada *Backpropagation*, Fase pelatihan dimulai dari proses *feed forward* yang digunakan untuk mencari nilai hidden layer dan nilai output layer, jika hasil output layer tidak sesuai dengan target maka dilakukan proses *backward* untuk mencari nilai *error* yang ada pada output layer dan hidden layer. Dan nantinya akan mempengaruhi nilai bobot yang ada pada hidden layer dan input layer setelah itu dilakukan kembali *feedforward* dan *backward* jika hasil *output* tidak sesuai dengan target. Arsitektur *Backpropagation* Braille yaitu input sebanyak 6 unit data, lapisan hidden sebanyak 10 unit dan output sebanyak 5 unit.

Gambar 4 menunjukkan arsitektur JST *Backpropagation* untuk pengenalan karakter Braille.



**Gambar 4.** Arsitektur *Backpropagation*

Proses pelatihan jaringan saraf tiruan ini menggunakan pasangan data. Table 2 adalah pasangan data yang digunakan untuk pelatihan *Backpropagation*.

**Tabel 2.** Pasangan Data *Backpropagation*

No.	Huruf Braille	Pasangan Data				
		Y0	Y1	Y2	Y3	Y4
1	a	0	0	0	0	1
2	b	0	0	0	1	0
3	c	0	0	0	1	1
4	d	0	0	1	0	0
5	e	0	0	1	0	1
6	f	0	0	1	1	0
7	g	0	0	1	1	1

8	h	0	1	0	0	0
9	i	0	1	0	0	1
10	j	0	1	0	1	0
11	k	0	1	0	1	1
12	l	0	1	1	0	0
13	m	0	1	1	0	1
14	n	0	1	1	1	0
15	o	0	1	1	1	1
16	p	1	0	0	0	0
17	q	1	0	0	0	1
18	r	1	0	0	1	0
19	s	1	0	0	1	1
20	t	1	0	1	0	0
21	u	1	0	1	0	1
22	v	1	0	1	1	0
23	w	1	0	1	1	1
24	x	1	1	0	0	0
25	y	1	1	0	0	1
26	z	1	1	0	1	0

#### 2.4. Pengenalan Karakter Braille

Pada proses pengenalan menggunakan JST *Backpropagation* ini, dilakukan pencocokan antara data fitur yang diperoleh dari proses *thresholding* dengan pasangan data pelatihan. Dengan metode tersebut, proses pengenalan hanya menggunakan fase propagasi maju (*feedforward*) saja. Hal tersebut dikarenakan keempat data – data bobot yang dihasilkan pada proses *learning* sebelumnya sudah didapatkan dan telah disimpan di dalam file .txt, sehingga hanya memadukan antara nilai data yang telah diperoleh dari proses ekstraksi fitur karakter Braille sebagai data input dengan pasangan data yang telah ditentukan sebelumnya pada jaringan *output* atau target. Jadi proses pengenalan karakter Braille dapat dilakukan dengan baik dan benar.

#### 2.5. Pengujian JST *Backpropagation*

Jumlah hidden layer yang digunakan adalah sebanyak 10 hidden. Jumlah input pada layer input sebanyak 6 input data, hal tersebut dikarenakan pada setiap titik atau dot pada karakter Braille memiliki sebanyak 6 kondisi data dan layer output atau target pada JST *Backpropagation* adalah sebanyak 5 data berupa data biner.

Sedangkan untuk jumlah pattern yang digunakan sebagai data pembelajaran JST *Backpropagation* ini adalah sebanyak 81 data *pattern*, karena dengan menggunakan data sebanyak 81 *pattern* tersebut maka pada sistem pengenalan karakter Braille ini dapat mengenali karakter braille dengan baik. Nilai toleransi *error* pada sistem ini juga sangat mempengaruhi saat

sistem melakukan proses pembelajaran. Karena semakin kecil nilai toleransi *error* yang diberikan pada sistem pembelajaran dengan JST *Backpropagation* ini, maka sistem akan semakin baik dalam proses pengenalan karakter Braille. Nilai toleransi *error* yang diberikan adalah sebesar 0.0001. Dengan nilai toleransi yang diberikan tersebut, sistem dapat mengenali karakter Braille dengan baik. Untuk nilai  $\mu$  nol pada sistem diberikan nilai sebesar 0.5 dan nilai K nol diberikan nilai sebesar 100.

#### 2.6. Pengujian Toleransi Error (error goal)

Proses pengujian ini dilakukan untuk menentukan besarnya toleransi *error* (*error goal*) yang menghasilkan tingkat keberhasilan tertinggi untuk pelatihan dan data pengujian dengan menggunakan unit tersembunyi sebanyak 10 unit. Penggunaan nilai toleransi *error* yang berbeda, diharapkan dapat meningkatkan nilai keberhasilan jaringan pada proses pengujian dengan data asing sehingga dapat mencapai nilai tingkat keberhasilan sebesar 100%. Entisari Rokhani[6], dalam penelitiannya menggunakan lima kombinasi nilai toleransi *error* yang berbeda yaitu 0.05, 0.005, 0.001, 0.0005, dan 0.0001.

**Tabel 3.** Hasil Percobaan Perubahan *error goal*

No	<i>Error Goal</i>	Iterasi	Dikenali / Tidak Dikenali	
			Data Pelatihan	Tingkat Keberhasilan (%)
1	0.05	71	81/7	74,07
2	0.005	1080	81/1	98,76
3	0.001	1583	81/0	100
4	0.0005	2314	81/0	100
5	0.0001	14929	81/0	100

Dari hasil percobaan toleransi *error* pada tabel 3 di atas menghasilkan tiga percobaan terbaik dengan nilai tingkat keberhasilan mencapai 100%. Yaitu pada *error goal* 0.001 dengan iterasi 1583, 0.0005 dengan iterasi 2314, dan 0.0001 dengan iterasi 14929. Berarti pada kombinasi parameter tersebut jaringan telah mampu mengenali semua pola karakter Braille yang diujikan dengan benar. Hasil tersebut merupakan hasil yang paling baik dan sesuai dengan tujuan simulasi jaringan bahwa jaringan mampu menghasilkan nilai tingkat keberhasilan sebesar 100% untuk semua data pelatihan.

#### 2.7. Pengujian Variasi Miu Nol

Percobaan berikut menggunakan nilai toleransi *error* terbaik yang didapat dari percobaan sebelumnya dan menggunakan nilai  $\mu$  nol yang berbeda

(0.1,0.2,0.5,0.8,0.9,1) untuk dianalisa hasil terbaik berdasarkan nilai *miu nol* tersebut. Pada pengujian ini akan dilihat pengaruh dari nilai *miu nol* terhadap tingkat keberhasilan pengenalan dan jumlah iterasi untuk menghasilkan *output* pada JST *Backpropagation*. Pengujian ini dilakukan dengan memberikan nilai toleransi *error* yang sama yaitu 0.001 dan nilai *miu nol* yang berubah – ubah. Tabel 4 menunjukkan hasil pengaruh *miu nol* :

**Tabel 4.** Hasil Percobaan *Miu Nol*

No.	Miu Nol	Iterasi	MSE	Prosentasi Kesalahan (%)
1	0.1	1999	0.0269	96
2	0.2	1999	0.0095	89
3	0.5	1583	0.0009	0
4	0.8	1999	0.0372	97
5	0.9	1999	0.0141	92
6	1	1999	0.0692	98

Dari hasil percobaan yang dihasilkan seperti pada tabel 4 di atas, bahwa nilai *miu nol* terbaik untuk proses pengenalan dan sesuai dengan tujuan simulasi dari JST *Backpropagation* adalah dengan nilai *miu nol* sebesar 0.5. Dimana nilai toleransi *error*nya sama dengan nilai toleransi yang telah ditetapkan sebelumnya yaitu 0.001 dan prosentasi kesalahan yang dihasilkan 0% atau diperoleh nilai keberhasilan 100% pada proses iterasi 1583, yang menunjukkan bahwa pada jaringan *Backpropagation* dapat belajar dengan baik pada kondisi ini. Oleh karenanya kombinasi nilai toleransi *error* dan nilai *miu nol* inilah yang dapat dijadikan acuan terbaik untuk proses pelatihan jaringan.

## 2.8. Pengujian Pengenalan Karakter Braille

Jaringan Saraf Tiruan *Backpropagation* didesain dan dilatih untuk mengenali 26 karakter Braille. Dari huruf Braille “a” sampai huruf Braille “z” metode JST *Backpropagation* bisa mengenali ke 26 huruf tersebut dengan benar. Arsitektur terbaik jaringan saraf tiruan dengan algoritma *Backpropagation* untuk mengenali pola karakter Braille ditentukan dengan cara melakukan beberapa percobaan dengan setiap kombinasi yang berbeda, mulai dari pengujian toleransi *error* (*error goal*) sampai pada pengujian nilai *miu nol*nya, dengan melakukan beberapa pengujian tersebut sehingga bisa dihasilkan pengenalan karakter Braille dengan baik. Dari percobaan – percobaan tersebut dipilih hasil terbaik yang ditentukan berdasarkan nilai tingkat keberhasilan tertinggi untuk data pelatihan jaringan. Dimana nilai tertinggi tersebut diharapkan dapat mencapai 100% dengan asumsi bahwa jaringan mampu mengingat dan dapat mengenali

semua pola karakter Braille yang pernah dilatih. Hasil pengujian data dengan JST *Backpropagation* pengenalan karakter Braille dapat dilihat seperti pada table 5 berikut :

**Tabel 5.** Hasil Pengenalan Semua Pola karakter Braille

No.	Huruf Braille	Input Data	Output
1	a	100000	a
2	b	110000	b
3	c	100100	c
4	d	100110	d
5	e	100010	e
6	f	110100	f
7	g	110110	g
8	h	100110	h
9	i	010100	i
10	j	010110	j
11	k	101000	k
12	l	111000	l
13	m	101100	m
14	n	101110	n
15	o	101010	o
16	p	111100	p
17	q	111110	q
18	r	111010	r
19	s	011100	s
20	t	011110	t
21	u	101001	u
22	v	111001	v
23	w	010111	w
24	x	101101	x
25	y	101111	y
26	z	101011	z

## 3. Penutup

Aplikasi Jaringan Saraf Tiruan dengan algoritma *Backpropagation* terbaik telah dihasilkan untuk mengenali semua karakter Braille dengan arsitektur yang terdiri dari 6 unit input, 10 unit pada lapisan tersembunyi, dan 5 unit lapisan output atau unit target. Serta parameter – parameter pelatihan, yaitu toleransi *error* (*error goal*) sebesar 0.001, dan fungsi aktivasi setiap lapisan adalah sigmoid biner. Tingkat keberhasilan yang dihasilkan dari arsitektur jaringan saraf tiruan dengan algoritma *Backpropagation* tersebut mencapai 100% untuk masing – masing data pelatihan, dengan *pattern* sebanyak 81 data dan *training* berhenti pada epoch ke 651.

## Daftar Pustaka

- [1]. Didi Tarsidi, “Pedoman Format Braille”, Makalah Jurusan Pendidikan Luar Biasa Universitas Pendidikan Indonesia (UPI) Bandung, 2000.
- [2]. Muhammad Andy Andiansyah, “Pengembangan

*Teknologi Komputer & Informasi sebagai Langkah Rehabilitasi Tuna Netra di Indonesia*”, Teknik Riset Operasi.

- [3]. Achmad Basuki, Nana Ramadijanti, Riyanto Sigit, “*Modul Ajar Praktikum Pengolahan Citra*”, Politeknik Elektronika Negeri Surabaya, 2008.
- [4]. Thiang, “*Pengenalan Huruf Braille dengan Menggunakan Hamming Network*”, Jurusan Teknik Elektro Universitas Kristen Petra Surabaya.
- [5]. Endah Sri Utami, “*Pembacaan Plat Kendaraan Menggunakan Jaringan Saraf Tiruan (JST) Backpropagation Berbasis Image Processing*”, Proyek Akhir Politeknik Elektronika Negeri Surabaya, 2009.
- [6]. Entisari Rokhani, “*Pengenalan Pola Huruf Tulisan Tangan Menggunakan Jaringan Saraf Tiruan dengan Algoritma Backpropagation*”, Skripsi Jurusan Pendidikan Matematika Universitas Negeri Yogyakarta, 2009.
- [7]. F. Wauquier, J.P. Dubus, M. Benjelloun, P. Altmayer, V. Devlaminck, “*Image Processing Techniques to Perform An Autonomous System to Translate Relief Braille Into Black-ink, Called: LectoBraille*”, IEEE Engineering in Medicine and Biology Society, 1988.
- [8]. Y.M. Alginahi, “*Thesholding and Character Recognition In Security Documents With Watremarked Background*”, IEEE, 2008.
- [9]. Lisa Wong, Stephan Hussmann, Waleed Abdulla, “*A Software Algorithm Prototype for Optical Recognition of Embossed Braille*”, Dept. of Electrical & Computer Engineering.
- [10]. Asep Sholahuddin, MT, “*Penerapan Neural Network Tentang Metode Backpropagation Pada Pengenalan Pola Huruf*”, Jurusan Matematika Universitas Padjadjaran.